

CLAIMS

What is claimed is:

- 1 1. A method of software change modeling for nodes in a distributed network of nodes,
2 the method comprising the computer-implemented steps of:
3 providing a master node;
4 receiving a software update for a node on said master node;
5 wherein the software update contains a software package or a set of software
6 packages;
7 wherein a software package contains at least one software module with corresponding
8 software dependency information;
9 wherein said master node notifies said node that a software update is being requested;
10 and
11 wherein said master node passes said node identities of software package(s) to be
12 updated and software dependency information.
- 1 2. A method as recited in Claim 1, wherein said node determines running processes on
2 said node that will be affected by the software update using the software dependency
3 information.
- 1 3. A method as recited in Claim 2, wherein said node notifies processes that have
2 indicated interest in software updates that the software update is being requested; wherein
3 each notified process evaluates the effect that the software update will have on its operation;
4 wherein if any of the processes determine that the software update will degrade or have a
5 negative impact on said node's normal operation, the process returns a veto to said node
6 along with reasons why; and wherein if a process finds that the software update will have no
7 negative effects, the process returns an acceptance of the software update to said node.

1 4. A method as recited in Claim 3, wherein said node waits for all of the notified
2 processes to return the results of their evaluations and once all of the processes have reported
3 to said node, said node notifies said master node if any of the processes have vetoed the
4 software update along with their reasons.

1 5. A method as recited in Claim 4, wherein said master node displays node identifiers
2 and the processes that have vetoed the software update along with their reasons to a user.

1 6. A method as recited in Claim 1, wherein a user initiates a software update by
2 installing an image containing the software update onto said master node.

1 7. A method as recited in Claim 6, wherein the user indicates what nodes and which
2 software package(s) are to be updated.

1 8. A method as recited in Claim 1, wherein a software package indicates the type of
2 node to which it applies.

1 9. A method as recited in Claim 1, wherein the software update contains a list of
2 software packages destined for each node.

1 10. A method as recited in Claim 1, wherein a software package contains version
2 information, dependency information, and other metadata information pertaining to software
3 in the package.

1 11. A method as recited in Claim 10, wherein the metadata includes a list of application
2 program interface (API) providers and consumers.

1 12. An apparatus of software change modeling for nodes in a distributed network of
2 nodes, comprising:
3 a master node;
4 means for receiving a software update for a node on said master node;
5 wherein the software update contains a software package or a set of software
6 packages;
7 wherein a software package contains at least one software module with corresponding
8 software dependency information;
9 wherein said master node notifies said node that a software update is being requested;
10 and
11 wherein said master node passes said node identities of software package(s) to be
12 updated and software dependency information.

1 13. An apparatus as recited in Claim 12, wherein said node determines running processes
2 on said node that will be affected by the software update using the software dependency
3 information.

1 14. An apparatus as recited in Claim 13, wherein said node notifies processes that have
2 indicated interest in software updates that the software update is being requested; wherein
3 each notified process evaluates the effect that the software update will have on its operation;
4 wherein if any of the processes determine that the software update will degrade or have a
5 negative impact on said node's normal operation, the process returns a veto to said node
6 along with reasons why; and wherein if a process finds that the software update will have no
7 negative effects, the process returns an acceptance of the software update to said node.

1 15. An apparatus as recited in Claim 14, wherein said node waits for all of the notified
2 processes to return the results of their evaluations and once all of the processes have reported
3 to said node, said node notifies said master node if any of the processes have vetoed the
4 software update along with their reasons.

1 16. An apparatus as recited in Claim 15, wherein said master node displays node
2 identifiers and the processes that have vetoed the software update along with their reasons to
3 a user.

1 17. An apparatus as recited in Claim 12, wherein a user initiates a software update by
2 installing an image containing the software update onto said master node.

1 18. An apparatus as recited in Claim 17, wherein the user indicates what nodes and which
2 software package(s) are to be updated.

1 19. An apparatus as recited in Claim 12, wherein a software package indicates the type of
2 node to which it applies.

1 20. An apparatus as recited in Claim 12, wherein the software update contains a list of
2 software packages destined for each node.

1 21. An apparatus as recited in Claim 12, wherein a software package contains version
2 information, dependency information, and other metadata information pertaining to software
3 in the package.

1 22. An apparatus as recited in Claim 21, wherein the metadata includes a list of
2 application program interface (API) providers and consumers.

1 23. A computer-readable medium carrying one or more sequences of instructions for
2 software change modeling for nodes in a distributed network of nodes, which instructions,
3 when executed by one or more processors, cause the one or more processors to carry out the
4 steps of:

5 providing a master node;
6 receiving a software update for a node on said master node;

7 wherein the software update contains a software package or a set of software
8 packages;
9 wherein a software package contains at least one software module with corresponding
10 software dependency information;
11 wherein said master node notifies said node that a software update is being requested;
12 and
13 wherein said master node passes said node identities of software package(s) to be
14 updated and software dependency information.

1 24. A computer-readable medium as recited in Claim 23, wherein said node determines
2 running processes on said node that will be affected by the software update using the
3 software dependency information.

1 25. A computer-readable medium as recited in Claim 24, wherein said node notifies
2 processes that have indicated interest in software updates that the software update is being
3 requested; wherein each notified process evaluates the effect that the software update will
4 have on its operation; wherein if any of the processes determine that the software update will
5 degrade or have a negative impact on said node's normal operation, the process returns a veto
6 to said node along with reasons why; and wherein if a process finds that the software update
7 will have no negative effects, the process returns an acceptance of the software update to said
8 node.

1 26. A computer-readable medium as recited in Claim 25, wherein said node waits for all
2 of the notified processes to return the results of their evaluations and once all of the processes
3 have reported to said node, said node notifies said master node if any of the processes have
4 vetoed the software update along with their reasons.

1 27. A computer-readable medium as recited in Claim 26, wherein said master node
2 displays node identifiers and the processes that have vetoed the software update along with
3 their reasons to a user.

1 28. A computer-readable medium as recited in Claim 23, wherein a user initiates a
2 software update by installing an image containing the software update onto said master node.

1 29. A computer-readable medium as recited in Claim 28, wherein the user indicates what
2 nodes and which software package(s) are to be updated.

1 30. A computer-readable medium as recited in Claim 23, wherein a software package
2 indicates the type of node to which it applies.

1 31. A computer-readable medium as recited in Claim 23, wherein the software update
2 contains a list of software packages destined for each node.

1 32. A computer-readable medium as recited in Claim 23, wherein a software package
2 contains version information, dependency information, and other metadata information
3 pertaining to software in the package.

1 33. A computer-readable medium as recited in Claim 32, wherein the metadata includes a
2 list of application program interface (API) providers and consumers.

1 34. A method of software change modeling of networked nodes on a computer system,
2 the method comprising the computer-implemented steps of:
3 providing a software update simulator on said computer system;
4 simulating processes from at least one node on said computer system;
5 wherein each functional process from said node is a minimal version of a functional
6 process that runs on said node and
7 receiving a software update for a node by said software update simulator;
8 wherein the software update contains a software package or a set of software
9 packages;

10 wherein a software package contains at least one software module with corresponding
11 software dependency information;
12 wherein said software update simulator notifies a control process for said node that a
13 software update is being requested; and
14 wherein said software update simulator passes said control process identities of
15 software package(s) to be updated and software dependency information.

1 35. A method as recited in Claim 34, wherein said control process determines running
2 functional node processes that will be affected by the software update using the software
3 dependency information.

1 36. A method as recited in Claim 35, wherein said control process notifies processes that
2 have indicated interest in software updates that the software update is being requested;
3 wherein each notified process evaluates the effect that the software update will have on its
4 operation; wherein if any of the processes determine that the software update will degrade or
5 have a negative impact on said node's normal operation, the process returns a veto to said
6 control process along with reasons why; and wherein if a process finds that the software
7 update will have no negative effects, the process returns an acceptance of the software update
8 to said control process.

1 37. A method as recited in Claim 36, wherein said control process waits for all of the
2 notified processes to return the results of their evaluations and once all of the processes have
3 reported to said control process, said control process notifies said software update simulator
4 if any of the processes have vetoed the software update along with their reasons.

1 38. A method as recited in Claim 37, wherein said software update simulator displays
2 node identifiers and the processes that have vetoed the software update along with their
3 reasons to the user.

1 39. A method as recited in Claim 34, wherein a user initiates a software update by
2 loading an image containing the software update into said software update simulator.

1 40. A method as recited in Claim 39, wherein the user indicates what nodes and which
2 software package(s) are to be updated.

1 41. A method as recited in Claim 34, wherein a software package contains version
2 information, dependency information, and other metadata information pertaining to software
3 in the package.

1 42. A method as recited in Claim 41, wherein the metadata includes a list of application
2 program interface (API) providers and consumers.

1 43. A method of software change modeling of nodes in a network of nodes on a computer
2 system, the method comprising the computer-implemented steps of:
3 providing a software update simulator on said computer system;
4 wherein said software simulator runs software components normally run on a master
5 node in the network of nodes;
6 wherein a user loads a node's current software configuration into said software
7 simulator by loading current software modules installed on a node;
8 wherein the user requests a simulation of a software update by loading an updated
9 software image into said simulator;
10 wherein the software image contains a software package or a set of software
11 packages;
12 wherein a software package contains at least one software module with corresponding
13 software dependency information;
14 wherein said software simulator calculates the software update's impact on said node
15 using the current software configuration of said node; and
16 displaying the calculation's results to the user.

1 44. A method as recited in Claim 43, wherein the user also indicates to said software
2 simulator the type of node being analyzed.

1 45. A method as recited in Claim 43, wherein said software update is a software
2 downgrade where modules are being removed.

1 46. An apparatus of software change modeling of nodes in a network of nodes on a
2 computer system, comprising:
3 a software update simulator on said computer system;
4 wherein said software simulator runs software components normally run on a master
5 node in the network of nodes;
6 wherein a user loads a node's current software configuration into said software
7 simulator by loading current software modules installed on a node;
8 wherein the user requests a simulation of a software update by loading an updated
9 software image into said simulator; and
10 wherein the software image contains a software package or a set of software
11 packages;
12 wherein a software package contains at least one software module with corresponding
13 software dependency information;
14 wherein said software simulator calculates the software update's impact on said node
15 using the current software configuration of said node; and
16 means for displaying the calculation's results to the user.

1 47. An apparatus as recited in Claim 46, wherein the user also indicates to said software
2 simulator the type of node being analyzed.

1 48. An apparatus as recited in Claim 46, wherein said software update is a software
2 downgrade where modules are being removed.

1 49. A computer-readable medium carrying one or more sequences of instructions for
2 software change modeling of nodes in a network of nodes on a computer system, which
3 instructions, when executed by one or more processors, cause the one or more processors to
4 carry out the steps of:
5 providing a software update simulator on said computer system;
6 wherein said software simulator runs software components normally run on a master
7 node in the network of nodes;
8 wherein a user loads a node's current software configuration into said software
9 simulator by loading current software modules installed on a node;
10 wherein the user requests a simulation of a software update by loading an updated
11 software image into said simulator;
12 wherein the software image contains a software package or a set of software
13 packages;
14 wherein a software package contains at least one software module with corresponding
15 software dependency information;
16 wherein said software simulator calculates the software update's impact on said node
17 using the current software configuration of said node; and
18 displaying the calculation's results to the user.

1 50. A computer-readable medium as recited in Claim 49, wherein the user also indicates
2 to said software simulator the type of node being analyzed.

1 51. A computer-readable medium as recited in Claim 49, wherein said software update is
2 a software downgrade where modules are being removed.